AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method in a computer system for scheduling tasks, the method comprising:

notifying a task executing on a parallel processor architecture having multiple simultaneously executing protection domains that ~~it~~ the task is being preempted from ~~processor utilization~~utilizing the processor;

in response to the notification, receiving an indication from the task that it is ready to be swapped out and an indication as to whether the task is blocked; ~~and~~

determining that the task can be swapped back in; and

when an indication that the task is blocked is received, deferring the swapping in of the task until an event occurs that may cause the task to become unblocked.

2. (Original) The method of claim 1 wherein the computer system is a multithreaded computer system.

3. (Original) The method of claim 1 wherein in response to the notification, the task saves its state.

4. (Original) The method of claim 1 wherein the event is an indication from an operating system.

5. (Original) The method of claim 1 wherein the indication of whether the task is blocked includes an identification of a thread of the task that is blocked.

6. (Original) The method of claim 1 including tracking a number of threads of the task that are blocked.

7. (Currently Amended) A system for scheduling tasks, the system comprising:

a component that notifies a task executing on a parallel processor architecture having multiple simultaneously executing protection domains that ~~it~~ the task is being preempted from ~~processor utilization~~utilizing the processor;

a component that, in response to the notification, receives an indication from the task that it is ready to be swapped out and an indication as to whether the task is blocked; ~~and~~

a component that determines that the task can be swapped back in; and

a component that, when an indication that the task is blocked is received, defers the swapping in of the task until an event occurs that may cause the task to become unblocked.

8. (Original) The system of claim 7 wherein the computer system is a multithreaded computer system.

9. (Original) The system of claim 7 wherein in response to the notification, the task saves its state.

10. (Original) The system of claim 7 wherein the event is an indication from an operating system.

11. (Original) The system of claim 7 wherein the indication of whether the task is blocked includes an identification of a thread of the task that is blocked.

12. (Original) The system of claim 7 including a component that tracks a number of threads of the task that are blocked.

13. (Currently Amended) A ~~method in a computer system~~computer-readable storage medium encoded with instructions for scheduling tasks, ~~the~~ by a method comprising:

executing instructions of a task on a parallel processor architecture having multiple simultaneously executing protection domains;

determining that a thread of ~~a~~ the task is blocked; and

sending an indication to an operating system that the task is ready to be swapped out and that the thread is blocked.

14. (Currently Amended) The computer-readable medium ~~method~~ of claim 13 wherein the determining is done by the task.

15. (Currently Amended) The computer-readable medium ~~method~~ of claim 13 including incrementing a variable relating to a number of blocked threads.

16. (Currently Amended) The computer-readable medium ~~method~~ of claim 13 including receiving an indication from the operating system that the thread is no longer blocked.

17. (Currently Amended) The computer-readable medium ~~method~~ of claim 16 including decrementing a variable relating to a number of blocked threads.

18. (New) The computer-readable medium of claim 13 wherein the task does not know which of the multiple protection domains on which it is executing.

19. (New) The method of claim 1 including when the task is swapped back in, executing the task within a different protection domain than the one on which it was originally executing.

20. (New) The method of claim 1 wherein the processor has 16 simultaneously active protection domains.